Multiscale contour description for pattern recognition

Kidiyo Kpalma, Joseph Ronsin

Pattern Recognition Letters 27 (2006) 1545–1559 (Preprint copy)

Multiscale contour description for pattern recognition

Kidiyo KPALMA* **and Joseph RONSIN**
IETR UMR–CNRS 6164, Groupe Image et Télédétection, INSA de Rennes
20, avenue des Buttes de Coësmes, CS 14315, 35043 RENNES CEDEX, France
{kidiyo.kpalma | joseph.ronsin}@insa-rennes.fr

Abstract

In this paper, we present a new method and its preliminary results within the context of pattern analysis and recognition. This method is based on the multiscale analysis of a curve and deals with the contour of planar objects. Our method uses a low-pass Gaussian kernel to gradually smooth the contour by decreasing the filter bandwidth. Applying gain control to the smoothed contour stretches it to the same scale as the original one so that both contours intersect. By varying the bandwidth and marking all the intersection points between the smoothed contour and the original one we can generate the Intersection Points Map (IPM) function.

The initial results obtained by applying this method to various contours appears to indicate that the IPM function has some very interesting properties within the context of pattern recognition. It is translation and rotation insensitive and also scale change resistant for a large range of scaling. The IPM function generated when applied to noisy contours shows that the method is resistant to noise for a range of noise energy.

Applying the features extracted by this method to retrieve a pattern from a database confirms the efficiency of the method.

Keywords

pattern analysis and recognition, multiscale smoothing, scale-space analysis, curvature zero-crossing, intersection point map

1 Introduction

Within the context of automatic pattern recognition, it is necessary to have a simple, fast and efficient algorithm especially when dealing with large databases. This algorithm must be resistant to rotation, translation, scale change and noise. Many techniques have been described in the literature related to this topic, however pattern feature extraction, which is the main problem in pattern recognition, is complex and an enormous amount of parameters are required to measure [1,4]: shape area, elongation, moment invariants, number of holes, colour and texture…

It is well known that objects, especially natural ones, consist of a more or less large range of scales; and that the aspect of the object can change from one scale to another. Faced with this situation, it is very difficult to significantly describe a pattern using only one meaningful scale. To overcome this problem, increasingly more pattern analysis (or pattern description) techniques are based on multiscale or multiresolution representation methods [9,14]. Within this context, methods based on the pattern itself [3,6,19,20] exist as well as methods based on pattern contour behaviour [7,8,10,12,13,15,16,17,18].

This study deals exclusively with pattern contour methods. In this paper, we present the initial results of a new method (MSGPR: A Multi-Scale curve smoothing for Generalised Pattern Recognition) [2] within the context of pattern analysis for pattern recognition. This scale-space [7,8,12,13,14,15,16] method is based on multiscale smoothing of a planar pattern contour. According to its framework, this method is totally translation insensitive. As we showed in the initial studies [2], the MSGPR method is also rotation insensitive and robust against scale change for a large range of scaling.

More sophisticated methods exist that deal with local invariant features, associated with planar curve smoothing: this enables to recognise occulted patterns by analysing the non-occulted part of their contour [16]. The main weakness of the MSGPR approach is that it fails to handle occulted contours and contours having undergone a non-rigid deformation. Nevertheless, when concerned by non-occulted planar contours, MSGPR method is advantageous in that it is simple and easy to apply.

* Corresponding author. Tel.: +33 2 23 23 86 59; Fax.: +33 2 23 23 82 62 ; email: kidiyo.kpalma@insa-rennes.fr

The preliminary results indicate that the method is also resistant to additive noise.

By comparing pattern retrieval results obtained with this method with those provided by an adapted-CSS method, we are able to say that our method brings a positive point in the description of planar forms.

After this introduction, in the next section, we recall the method structure. In section 3, we present some of the main properties of the IPM function and we then go on to propose some interesting attributes that can be extracted from the IPM function for pattern characterisation. Section 4 concentrates on the application results within the context of pattern recognition by using the proposed attributes. Finally, in section 5, we present our conclusions together with some discussions.

2 Description of the MSGPR method

The framework of this method can be broken down into four main stages as follows (see Fig. 1):

1. The input contour is separated into two parameterised functions,
2. Both functions are low-pass filtered (smoothed),
3. Gain control is then applied to both filtered functions so that the corresponding smoothed contour has the same scale as the input one,
4. Finally, the intersection points map (IPM) is constructed by detecting the intersection points of the input contour and the smoothed gain-controlled one.

2.1 Pre-processing

Before applying the MSGPR method, the input contour is normalised in relation to the contour length. This pre-processing method consists of the following three tasks:

- Contour resampling. The contour is resampled so that there are 360 equally spaced points. This is achieved by applying linear interpolation to the original contour points: if the number of points is greater than 360, this corresponds to sub-sampling. Using 360 equally spaced points produces points with a space of 1° between two consecutives points, in the circle limit case: of course, any other number of points can be chosen.
- Scale changing. Since the input contour points are pixels, their coordinates must be integer numbers. We therefore apply a scale factor to the resampled contour to generate a contour with 360 equally spaced points with approximately one unit distance between two consecutive points. The coordinates of these points are rounded up to integer values.
  After this pre-processing, the subsequent operations are real-valued so the results are no more rounded up to integer values.
- Rotation compensation so that the major principal axis is aligned with the x-axis and the point furthest from the centre of gravity is on the right: the centre of this rotation is set to the centre of gravity of the contour. This defines the P_0 starting point for subsequent processing (see Fig. Fig. 2).

NB: In the following sections, the expression "input contour" will be used to describe the pre-processed contour.

2.2 Coordinate separation

The input contour is represented by a series of points defined by their (x,y) coordinates. First, the input contour is separated into two functions x(u) and y(u) which are functions of the normalised curvilinear u parameter that varies from 0 to 2π relative to the curve length (see Fig. 3). Each point of the curve is then represented by its coordinates $(x(u), y(u))$.

For each contour, the origin of the u parameter is set to the P_0 starting point on the x-axis as defined previously. The points are then reorganised in a counter clockwise direction as illustrated in Fig. 3. Since the contour is a closed curve, the two functions are periodic with period $T=2\pi$ relative to the u parameter.

2.3 Curve smoothing

Functions x(u) and y(u) are then gradually smoothed by decreasing the filter bandwidth. Similarly to the curvature scale space (CSS) method [7,12,15] or other scale-space methods and according to the arguments stated in [14], smoothing is based on the Gaussian filters $g(\sigma,u)$:

$$g(\sigma,u)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{u^2}{2\sigma^2}} \tag{1}$$

where σ is the standard deviation of the Gaussian kernel. The frequency response of the filter is then given by $G(\sigma, f)$

$$G(\sigma,f)=e^{-2\pi^2\sigma^2f^2} \tag{2}$$

where f denotes the frequency variable. The –3dB cut-off frequency of this filter is $f_c=\beta/\sigma$ where $\beta=\sqrt{Ln(2)}/(2\pi)=0.1325$ and its bandwidth which equals $B=2f_c=2\beta/\sigma$ is inversely proportional to the standard deviation σ.

The filtered functions are then given by: $X(\sigma,u)=g(\sigma,u)*x(u)$ and $Y(\sigma,u)=g(\sigma,u)*y(u)$ so that each $(x(u),y(u))$ point on the input contour leads to the $(X(\sigma,u),Y(\sigma,u))$ point on the output smoothed contour.

Since the bandwidth is conversely proportional to σ, it is clear that the bandwidth decreases as σ increases. Thus the filter cuts increasingly lower so that the output functions move towards their mean values when σ tends towards infinity. Fig. 4.a shows the filtered functions for σ=10 and σ=180 with their corresponding smoothed contours. Due to this filtering process, the $X(\sigma,u)$ and $Y(\sigma,u)$ functions are attenuated so that the reconstructed smoothed contour does not have the same scale as the original one (see Fig. 4.a).

2.4 Gain control

After low-pass filtering, the gain control system stretches the output contour so that it reaches the same scale as the input one and so that both curves intersect at certain points. The smoothed gain-controlled contour is then given by the coordinates $(X_{GC}(\sigma,u),\ Y_{GC}(\sigma,u))$ defined by

$$X_{GC}(\sigma,u)=A_x\left[X(\sigma,u)-X_G(\sigma)\right]+X_G(\sigma) \tag{3}$$

$$Y_{GC}(\sigma,u)=A_y\left[Y(\sigma,u)-Y_G(\sigma)\right]+Y_G(\sigma) \tag{4}$$

where $(X_G(\sigma),\ Y_G(\sigma))$ are the coordinates of the centre of gravity of the smoothed contour. The (A_x and A_y) gain control coefficients can be determined in different ways and we can look out for the optimal one. For the results presented in this paper, A_x and A_y coefficients are defined as follows:

$$A_x=\frac{\sum_{n=1}^{N}\left|x(u_n)-x_G\right|}{\sum_{n=1}^{N}\left|X(\sigma,u_n)-X_G(\sigma)\right|}\ \text{and}\ A_y=\frac{\sum_{n=1}^{N}\left|y(u_n)-y_G\right|}{\sum_{n=1}^{N}\left|Y(\sigma,u_n)-Y_G(\sigma)\right|} \tag{5}$$

where $\{u_n\}$ are the N (N=360) samples of the u parameter.

Fig. 4.b shows the smoothed contour after the gain control system. The input and smoothed gain-controlled contours are now on the same scale so that they can intersect.

By increasing σ, the output gain-controlled contour moves towards a convex curve and has some intersection points with the input contour. By marking these intersection points for each σ, we obtain the intersection points map (IPM) function defined below which characterises the pattern.

2.5 Definition of the IPM function

After the gain control system, the IPM function is generated as follows. For each σ value, we define a function which is an image in the (u,σ) plane so that (see Fig. 5):

- $IPM(u,\sigma)=0$ (black) if the $(x(u),y(u))$ point is an intersection point between the original curve and the filtered gain-controlled one,
- $IPM(u,\sigma)=1$ (white) if point $(x(u),y(u))$ is not an intersection point.

Fig. 5.a shows the input and the smoothed gain-controlled one (that corresponds to σ=180). This figure also

represents the IPM points for the corresponding σ. As σ increases, the number of IPM points decreases and we obtain "vertically" oriented lines or bands with variable widths. These remaining points constitute the points referred to as "limit IPM points". On Fig. 5.b, we represent the IPM function for σ, varying from 0 to 180, the IPM points indicated by (1) to (6) refer to the corresponding intersection points on Fig. 5.a. Notice that for small σ values corresponding to large bandwidth filters, most of the contour points are IPM points.

By analysing Fig. 5 and Fig. 6, it appears that low values of σ produce fine details on a contour and high values produce coarse structures. In this way, we can say that high frequency information lies in the lower part of the IPM function since low frequency behaviour is found in its higher part. In other words, we can say that the IPM function contains all the information concerning the pattern from finer scales (details) to coarser scales. As σ increases, details are eliminated from the contour which becomes smoother and smoother so that the remaining intersection points can reasonably be expected to be significantly representative of the pattern being studied.

We also notice that the concave and convex parts of a contour are characteristic of the IPM function as illustrated on Fig. 6. In this figure, the convex parts labelled (a), (b) and (c) on the contour lead to the structures labelled (A), (B) and (C), respectively, on the IPM function.

3 Pattern characterisation

To show that this function is characteristic of the contour being studied, Fig. 7 presents two other patterns and their IPM functions. Fig. 7.c represents the IPM function of the letter "M" (Fig. 7.a) and Fig. 7.d shows the IPM function of the letter "W" (Fig. 7.b). As can be seen, these IPM functions are, somewhat, topologically similar. However, in terms of the distance between two consecutive IPM points at scale σ, we notice that these IPM functions are different from each another and also different from those presented in Fig. 5. This shows that the IPM function is a good candidate to generate features for pattern recognition.

In the following sections, we will observe the behaviour of the IPM function regarding transformations such as scale change and rotation. We will then propose some features that can be derived from the IPM function.

3.1 The IPM function and scaling

In this section, we present the behaviour of the IPM function in relation of the scaling. The scaling transformation is obtained by applying a multiplicative coefficient to the contour coordinates. Fig. 8 shows that the IPM function is scale insensitive for a large range of scales. In this figure, the multiplicative coefficient next to the IPM function represents the scaling factor applied to the corresponding contour. According to the initial results obtained by applying the MSGPR method to the contour, the amount of limit IPM points remains constant and the IPM function does not change. For scale factors higher than the unit, the (entire) IPM function remains unchanged.

3.2 The IPM function and rotation

Fig. 9 shows a pattern (see Fig. 9.a) and its rotated copies (see Fig. 9.b through Fig. 9.f). The rotation angles are 60°, 120°, 180°, 240° and 300° counter-clockwise. As can be seen in these figures, all IPM functions are topologically identical, except for a "circular" delay. In this way, rotating the contour around its centre of gravity is equivalent to a circular delay in the IPM function. By analysing the IPM functions obtained from a contour and its rotated copy, we can directly read the rotation angle between both patterns. This is illustrated in Fig. 10. Since the u parameter is normalised to 2π, we can locate an IPM point A in one IPM function, determine its u_0 abscissa and then locate the corresponding IPM point A' in the second IPM function and also determine its abscissa u_1. The γ rotation angle between both contours is then determined by the difference between both abscissa: $\gamma = u_1 - u_0$. Since the pre-processing method aligned the input contour with its major principal axis, the information related to the orientation is not available for subsequent processing. Therefore, to be able to directly read the rotation angle, we need to keep the compensated angle in mind during the pre-processing stage.

3.3 Feature definition and selection

With this IPM function, we can predetermine a value for σ and use the number N(σ) of IPM points as characterisation features. Since the number of IPM points is circular-delay insensitive, the two patterns in

Fig. 7 are likely to be identical. To differentiate them, we need to use certain other features such as those suggested below.

By analysing the IPM functions obtained earlier, we notice that certain measurements remain invariant regardless of the transformation. These measurements can be used to characterise a pattern for recognition purposes. At present, we can suggest some heuristic features such as:

- The number of IPM limits points (those corresponding to the highest value of σ),
- The set of distances between two consecutive IPM points at a given value of σ,
- The minimum and maximum distances between two consecutive IPM limit points,
- Measurements related to convex/concave parts of irregular objects. As mentioned earlier (see Fig. 6), the IPM function corresponding to irregular objects produces certain particular noticeable structures that can produce some interesting features such as their height, area and width at a fixed value of σ,
- The ratio of arclength of arcs between two consecutive IPM points on the original contour over the Cartesian distance between both points for a given value of σ.

4 Application of the MSGPR method to pattern recognition

To evaluate the IPM function within the context of pattern recognition, we apply it to a pattern from a database. This database consists of 1,167 different contours plus 100 transformed copies of the same contour (kk707) and 25 transformed copies of 25 other contours. Transformations applied to the contours are affine transformations which are a combination of a rotation around the centre of gravity and scaling. Table I shows a set of ten ($R_{i=0,1,\ldots,9}$) rotation angles and a set of ten ($H_{i=0,1,\ldots,9}$) scaling factors. With these references, RmHnXYZ is obtained after applying an R_m rotation angle and an H_n scaling factor to the XYZ pattern. By way of example (see Table I), the R7H3kk707 pattern is generated by applying a 270° rotation angle and a 0.75 scaling factor to the kk707 contour of the database. Fig. 11 provides a sample of the contours contained in our test database.

Note that during the transformation procedure, the output point coordinates are all integer values. This enables us to deal with points as if they were pixels produced by edge detection in a bitmap image. Of course, as we will see below, this is rather unfavourable to our approach.

4.1 Attribute definition and selection

We can now describe and determine the features based on the IPM function that we used for our evaluation tests. To define these characterisation features, we first set σ to σ_0 value (e.g. $\sigma_0 = 180$). Then, for each pattern,

- We consider the IPM points (at the set σ_0 value of σ) and select two consecutive p_a and p_b points which are, circularly, the furthest apart in the IPM function as illustrated in Fig. 12,
- We determine the distance between both points to produce the first d_1 component of the V_0 attribute vector,
- The next components of V_0 are distances coming after d_1:

$$V_0 = (d_1, d_2, \ldots, d_{N-1}, d_N) \tag{6}$$

Thus, the V_0 attribute vector is of dimension N where N is the number of IPM points at scale σ_0. Using the most distant IPM points to define the attribute vector enables us to bypass the problem of correspondence between maxima of the interval tree as specified in [13].

To benefit from multi-scale information of the IPM function, we can define a set of M values of σ ($\sigma_0, \sigma_1, \ldots, \sigma_{M-1}$) and determine the V_i attribute vectors (i=0, 1, 2,…, M-1) corresponding to the σ_i scales (see equation 6). The V global attribute vector is then produced by a concatenation of the individual V_i scale vectors as follows:

$$V = (V_0, V_1, \ldots, V_{M-1}) \tag{7}$$

Each V_i vector is $N(\sigma_i)$ in length according to the number of IPM points on the σ_i scale. Thus, the length of the global attribute vector equals $N_T = \sum_{i=0}^{M-1} N(\sigma_i)$. Since V consists of vectors of multiple scales, it is considered to be a multiple scale attribute vector.

4.2 Similarity function

To measure the matching rate between two V_A and V_B attribute vectors associated to two different patterns, we define a similarity function as follows:

$$\mathrm{SimScore}(V_A, V_B) = 50(1+\cos(\gamma))\frac{\mathrm{Min}(\|V_A\|,\|V_B\|)}{\mathrm{Max}(\|V_A\|,\|V_B\|)} \tag{8}$$

where γ is the angle between both vectors and where $\|.\|$ indicates the module of a vector. This function ranges from 0% for very different vectors to 100% for perfectly matching vectors.

4.3 Results of pattern retrieval from a database

In this study, we used a set of five values for σ (180, 20, 30, 40, 50). Based on the total number of IPM points of the query contour that equals N_T=41, in the case of a query from the kk707 family, 72 contours are retained for subsequent recognition stages. In this situation, the dimension of the V global attribute vector is equal to N_T.

After the pre-selection process based on the total number of IPM points, we apply the SimScore function to all pairs composed with the Q_0 query and to each of the selected contours (see Table II). This pre-selection enables us to reduce computation time (at the SimScore function stage) by reducing the number of pairs to process.

Fig. 13 shows an example in which the Q_0 query pattern is a contour of the kk707 family i.e. a transformed copy of the kk707 pattern. This figure shows the 8 most similar contours to the queried contour. The similarity scores for these matching contours are 100%. Table II shows the 72 selected contours and their similarity scores in relation to the queried contour.

As can be seen in Table II and Fig. 13, out of the 72 remaining contours, the first 32 have a similarity score of 100%. It is interesting to note that, similarly to the query contour, they all come from the kk707 family. Another interesting point to note is that the highest similarity scores produced by a pattern from a different family is only 5%. The contour of the kk707 family with the lowest similarity score is the R6H1kk707 contour with a similarity score of 2%. This contour results from a (225°) rotation and a (*0.25*) scaling applied to the *kk707* contour: the low similarity score is due to the scale reduction applied to the original contour. Fig. 14 shows the R6H1kk707 contour which is expanded to the original size and placed on top of the original rotated contour (with a rotation angle of *225*°). This figure shows the defects introduced by the transformation applied to the original contour. These defects are due to scale reduction and mainly to the fact that the coordinates after transformation are rounded up to integer values (so that one point corresponds to one pixel in a bitmap image) as explained previously. This rounding up operation also introduces step-wise defects. These defects, which can produce peaks or self-intersections, are not dealt with in the present study. If these defects were taken into account, the results could be improved.

4.4 Comparison with an adaptation of CSS method

To evaluate our method, we have adapted the CSS (Curvature Scale Space) method. The maxima of the CSS curves and their position are then used like shape descriptor. Only maxima between the major maximum M and the maximum m=0.2M are taken into account to define the attribute vector. We then apply the similarity function of equation (8) after a pre-selection based on the length of the attribute vector as it is explained in section 4.3. To compare the retrieval results from both methods, we carried out four experiments :

1) The query shape "vase" is not in the database. The goal of this experiment if to assess the ability to discriminate shapes. This enables us to evaluate the sensitivity to false alarms situations in pattern recognition.
2) The query pattern "kk707" is in the database and there are also 100 members of the same family in the database. With this experiment, we aim at assessing the separability of the methods : how far are the less similar pattern from the query family and the most similar pattern from a different family (a non-parent pattern of the query).
3) We have applied both methods to a sample contours of the MPEG-Core Experiment CE-Shape-1. We applied them to retrieve a pattern from a database consisting of 20 contours of the same pattern that was subject of different kind of rigid or non-rigid transformations. Fig.18 shows an example of such contours : in this example, contours are extracted from a flying bat in different kind of motion.

From these experiments, one can note the following interesting points:

1) When the query doesn't exist in the database (see Fig. 15), the IPM based method retrieves 12 patterns with 26% maximal similarity score while the adapted CSS-based method retrieves 58 patterns with 50% maximal similarity score. Our method gives lower maximal similarity score (26%) than the CSS based method (50%). In this experiment, the lower the maximal similarity score is and the better is the recognition system: in deed when the similarity score is low, one can decide easily that the query pattern is unknown (doesn't exist in the database).
2) In the second experiment (see Fig. 16), the lowest similarity score obtained from a parent of the query pattern is 91% and the highest score obtained from a non-parent pattern is 43%: the difference between both scores is Δs_1=48%. With the adapted CSS method, this difference is Δs_2=1% while the lowest similarity score from a parent is 69%. Of course, a large difference indicates a good separation between classes: in other words, the best method is that method which provides large difference.
3) In the third experiment (see Fig. 17), when applied to contours of bats, the adapted CSS method yields better results than our method. As it can be seen on Fig. 17, some changes from a contour to another are not rigid transformations but they are caused by non-rigid deformations due to the motion of the animal. Since our method is essentially a global-driven method, it may fail for more or less non-rigid deformations. It is adapted better for rigid deformations as define for part A experiment in [17,18]. In this article, the authors have considered three different stages according to the three parts of the MPEG-7 Core Experiment CE-Shape-1: the part A that deals with robustness to scaling and to rotation is defined as a necessary condition to satisfy. In accordance with these experiments, we can reasonably say that our method is a good candidate to satisfy this necessary condition in shape description.

In any case, the query, itself, when it is present in the database, is retrieved with a score of 100%. As shown on Fig.18, bat-16, which is the query, is retrieved with a score of 100%. Bat-17 is retrieved with a similarity score of 89%: this high similarity score indicates that both patterns are similar as it can be seen visually on Fig.18. The next contour, bat-01, is found to be similar to the query with 40% similarity score. Visually, both contours are very similar but bat-01 has some high frequency details on the upper side of its wings : that explains the low score. Bat-05 and bat-04 present a very weak score (8% and 5%, respectively), because contrarily, with bat-01, their ears are quite apparent: what creates this strong dissimilarity with the query.

Table III summarises results that enable us to carry out a comparison of both the adapted CSS-based method and the MSGPR method. In this study, we used three families of contours: bats, bell and children extracted from the MPEG-7 Core Experiment CE-Shape-1 database. In each family, there are twenty contours representing different views or deformations due to the object movements. For a given family, we use each contour as a query and then record the minimal and maximal similarity score and the number of contours that are retained at the first step.

The average value of each of these three parameters is then registered in Table III. For each method, the average value of the maximal similarity score is 100% indicating that the query contour is always recognised with 100% similarity score. This parameter shows the capability of a method to recognise a pattern when this one exists in the database.

The average value of the minimal similarity score together with the average value of the number of retained patterns indicates the ability of the method to recognise a pattern that has undergone rigid or non-rigid deformations: the higher these parameters are and the better is the method. As can be seen on Table III, the two parameters seem to behave in opposition one of the other. While the IPM-based method provides the higher global average number of retained patterns and lower average of minimal similarity score, the CSS-based method provides the lower global average number of retained patterns and higher global average value of the minimum similarity score. That thus requires a compromise which will be guided by the contours under study.

5 Conclusion and discussion

We have presented some preliminary results of a new and simple method for planar pattern description within the context of pattern recognition. This paper analysis also the performance of some attributes derived from the IPM function. According to these results it would seem reasonable to believe that this is a good candidate method within the context of (planar) pattern recognition based on analysis of the closed non-occulted contour of the pattern being studied. This analysis method can be accelerated if just the IPM points for the concerned

set of σ values are generated rather than generating the "entire" IPM function. This is an advantage as opposed to the CSS method which has to construct the entire CSS map in order to generate the features.

Similarly to the CSS method, our approach has some interesting advantages:

- Curves do not have to be convex,
- The pattern can be identified regardless of its orientation (the method is rotation insensitive),
- The method is translation insensitive,
- The method is scale change insensitive for a large range of scales.

The initial results obtained by applying this method to pattern retrieving from a database show that the IPM function produces consistent attributes and can therefore be positively applied to pattern recognition, especially within the context of OCR (or Optical Character Recognition).

It is also interesting to note that the IPM function is very easy to understand and does not require much theory: this makes it very easy to implement. As stated above, our approach focuses on recognition of planar patterns that do not produce self-intersected contours. This aspect is therefore not taken into account in our studies.

Applying this method to noisy contours show that it is resistant to noise for a range of noise energy. From Fig. 19 we can note the following points: the IPM function is not too affected by noise for high scales (see upper part of the IPM functions); the apparition of many other structures in the lower part of the IPM function indicates that the pattern being studied is noisy. As stated in section 2.5, high frequency information is located in the lower part of the IPM function: as the added noise is represented by fine details (with high frequency), this explains the appearance of those structures for low scales.

As stated earlier, the main weakness of our approach is that it fails to handle occulted contours and those having undergone a non-rigid deformation, nevertheless it is advantageous in that it is simple and easy to to implement and to apply. It can also be considered as a useful step in a pattern recognition system, especially when one is concerned with rigid deformations.

Thanks to the results of the comparative study carried out between our method and an adapted-CSS method, we think that it is reasonable to say that our method has its own place in the context of planar shapes description and recognition.

Acknowledgements

We are grateful to Mr Sadegh Abasi, Dr Farzin Mokhtarian and Pr Josef Kittler for making their SQUID system publicly available.

References

[1] Zhang, D., and Lu, G., 2004. Review of shape representation and description techniques, Pattern Recognition, Vol.37, pp 1–19.

[2] Kpalma, K., and Ronsin, J., 2003. A Multi-Scale curve smoothing for Generalised Pattern Recognition (MSGPR), Seventh International Symposium on Signal Processing and its Applications (ISSPA), pp 427-430, Paris, France.

[3] Kadyrov A., Petrou, M., 2001. Object descriptors invariant to affine distortions. Proceedings of the British Machine Vision Conference, BMVC'2001, Manchester, UK.

[4] Sossa, H., 2000. Object Recognition, Summer School on Image and Robotics, INRIA Rhône-Alpes, France.

[5] Trimeche M., Alaya Cheikh F., and Gabbouj, M., 2000. Similarity Retrieval of Occluded Shapes Using Wavelet-Based Shape Feature, Proc. SPIE International Symposium on Internet Multimedia Management Systems (VV10), Boston, Massachusetts, USA.

[6] Torres-Méndez, L. A., Ruiz-Suárez, J. C., Sucar, L. E. and Gómez, G., 2000. Translation, Rotation, and Scale-Invariant Object Recognition, IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews, Vol. 30, No. 1, pp 125-130.

[7] Matusiak S., Daoudi M., Planar Closed Contour Representation by Invariant Under a General Affine Transformation, IEEE International Conference on System, Man and Cybernetics (IEEE-SMC'98), pp. 3251-3256, October 11-14, 1998, Hyatt Regency La Jolla, San Diego, California, USA.

[8] Wang Y.-P., Lee, S.L., and Toraichi, K., 1999. Multiscale curvature-based shape representation using B-spline wavelets, IEEE Transactions on Image Processing, Vol. 8, No 11, pp 1586-1592.

[9] Lindeberg, T., 1998. Principles for Automatic Scale Selection, Technical report ISRN KTH NA/P--98/14--SE. Department of Numerical Analysis and Computing Science, KTH (Royal Institute of Technology), S-100 44 Stockholm, Sweden.

[10] Roh, K.-S., Kweon, I.-S., 1998. 2-D object recognition using invariant contour descriptor and projective refinement, Pattern Recognition, Vol. 31, N° 4, pp. 441-455.

[11] A. Vapillon, B. Collin et A. Montanvert, Analyzing and Filtering Contour Deformation, International Conference on Image Processing (ICIP), Chicago, Illinois, USA, October 4-7, 1998.

[12] Bruckstein, A. M., Rivlin, E., and Weiss, I., 1996. Recognizing objects using scale space local invariants, Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96), August 25-29, pp. 760-764, Vienna, Austria.

[13] Mokhtarian, F., Abasi, S., and Kittler, J., 1996. Efficient and Robust Retrieval by Shape Content through Curvature Scale Space, in Proc. International Workshop on Image Databases and MultiMedia Search, pp 35-42, Amsterdam, The Netherlands.

[14] Lindeberg, T., 1994. Scale-Space Theory in Computer Vision, Kluwer Academic Publishers, Dordrecht, Netherlands.

[15] Mokhtarian, F., and Mackworth, A. K., 1992. A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves, in IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-14, N° 8.

[16] Bruckstein, A., Katzir, N., Lindenbaum, M., and Porat, M., Similarity invariant signatures for partially occluded planar shapes, IJCV, Vol. 7, No. 3, pp. 271-285, 1992.

[17] Latecki, L. J., Lakamper, R., and Eckhardt, U., Shape Dscriptors for Non-rigid Shapes with a Single Closed Contour, IEEE Conf. On Computer Vision and Pattern Recognition (CVPR), pp. 424-429, 2000

[18] Mokhtarian, F., and Bober, M., Curvature Scale Space Representation: Theory, Applications and MPEG-7 Standardization. Kluwer Academic, 2003.

[19] Grigorescu, C., and Petkov, N., Distance Sets for Shape Filters and Shape Recognition. IEEE Trans. Image Processing 12(9), 2003.

[20] Belongie, S., Malik, J., and Puzicha, J., Shape matching and object recognition using shape contexts. IEEE PAMI-24, No 24, pp 509-522, 2002

Figure captions

Fig.1

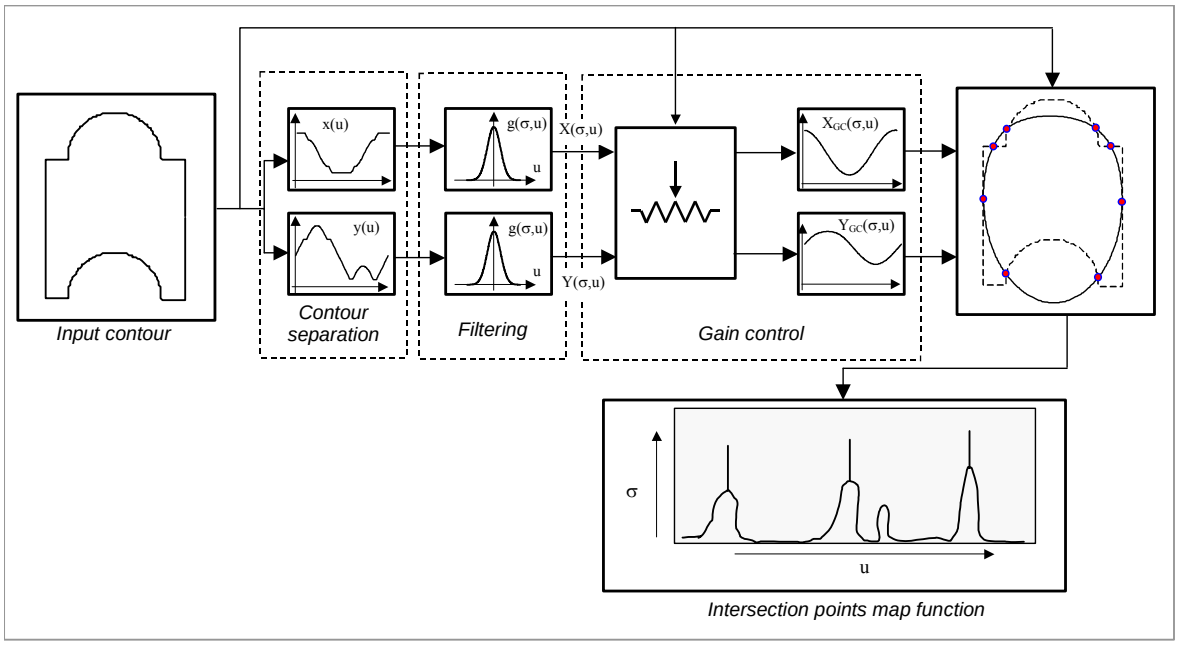

x(u)
y(u)
g(σ,u)
u
X(σ,u)
Y(σ,u)
X_GC(σ,u)
Y_GC(σ,u)
Input contour
Contour separation
Filtering
Gain control
σ
u
Intersection points map function

Fig.2

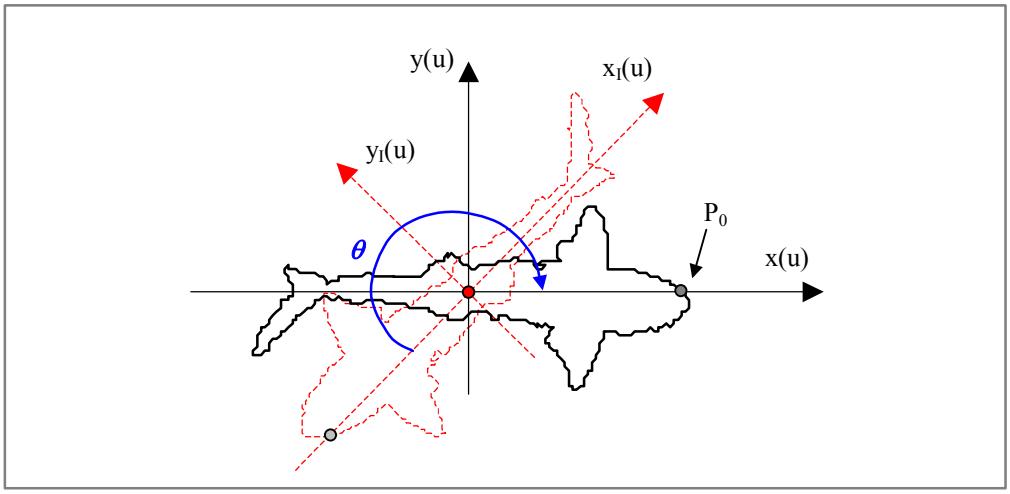


Fig.3

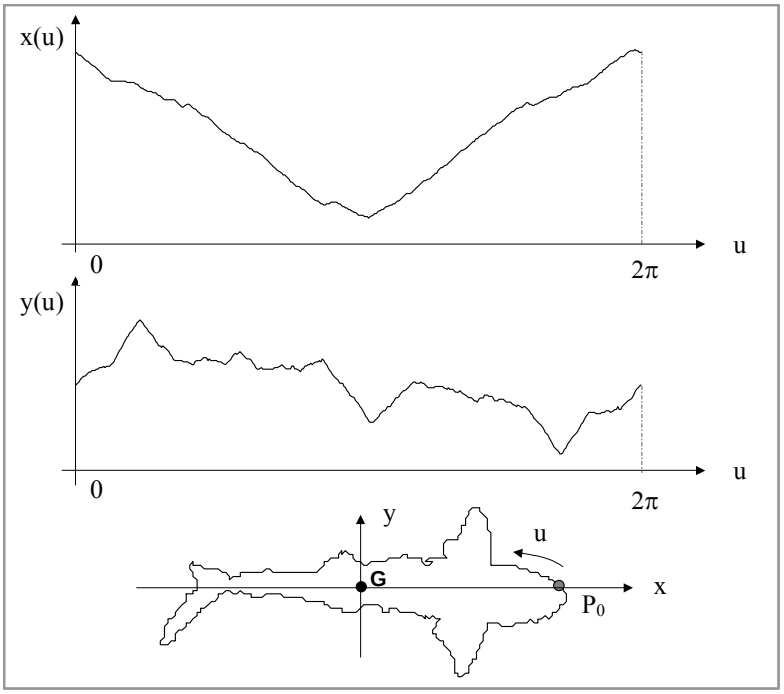


Fig.4

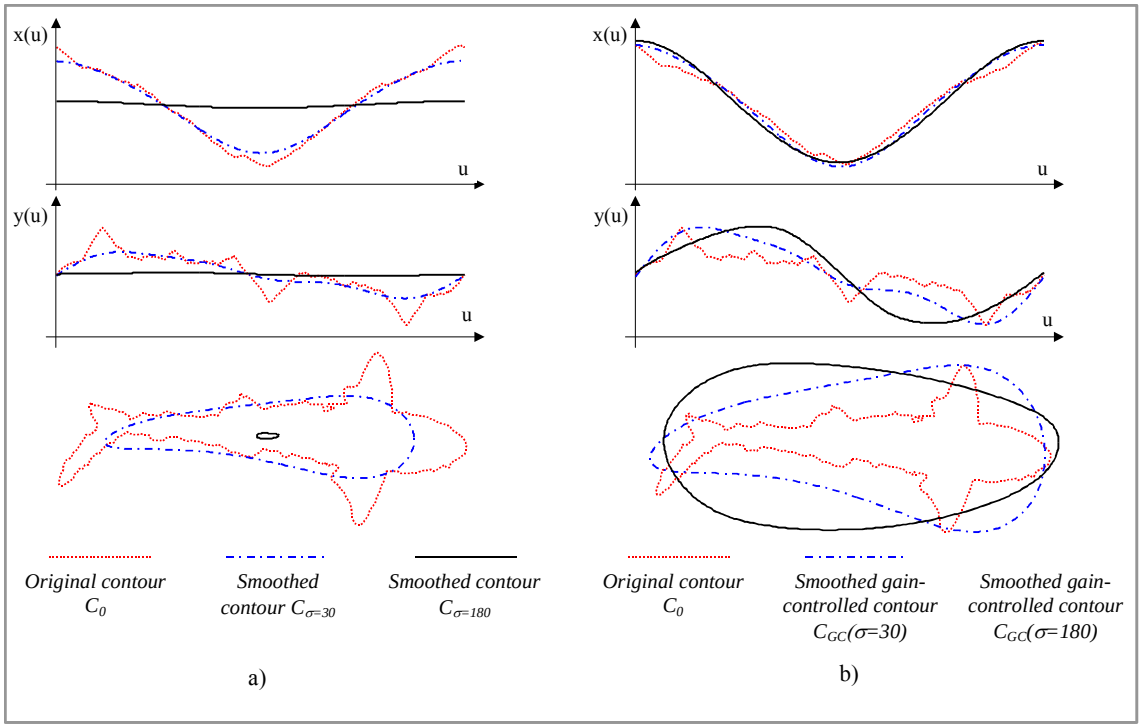

x(u)
u
y(u)
u
x(u)
u
y(u)
u
Original contour C_0
Smoothed contour $C_{\sigma=30}$
Smoothed contour $C_{\sigma=180}$
Original contour C_0
Smoothed gain-controlled contour $C_{GC}(\sigma=30)$
Smoothed gain-controlled contour $C_{GC}(\sigma=180)$
a)
b)

Fig.5

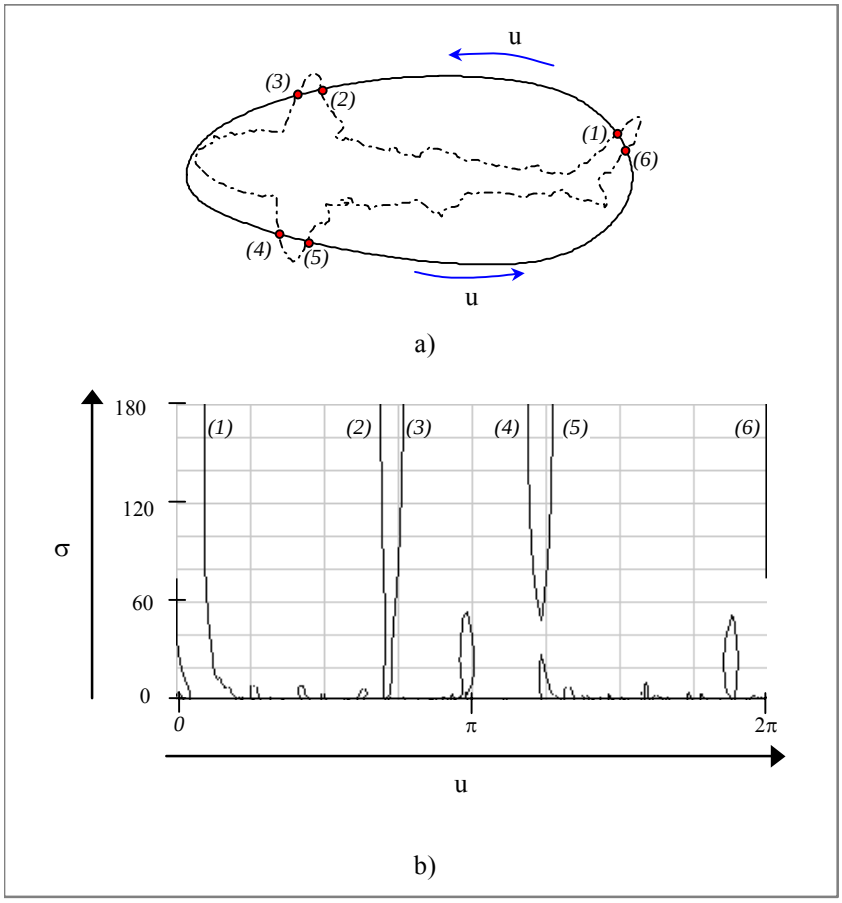


Fig.6:

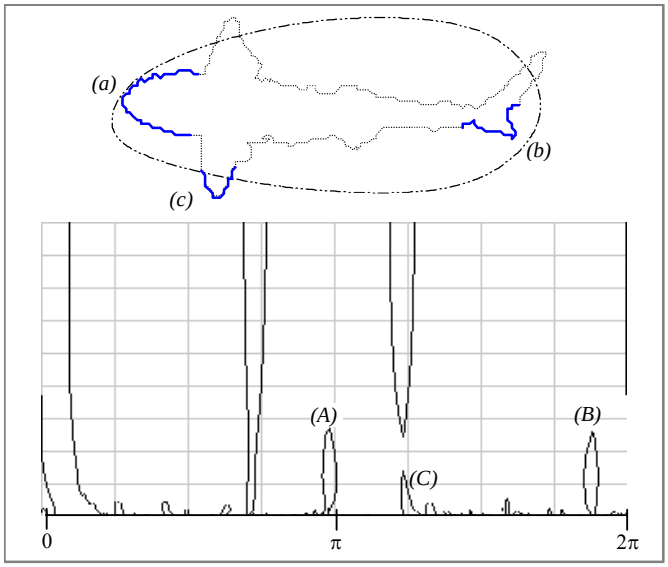


Fig.7:

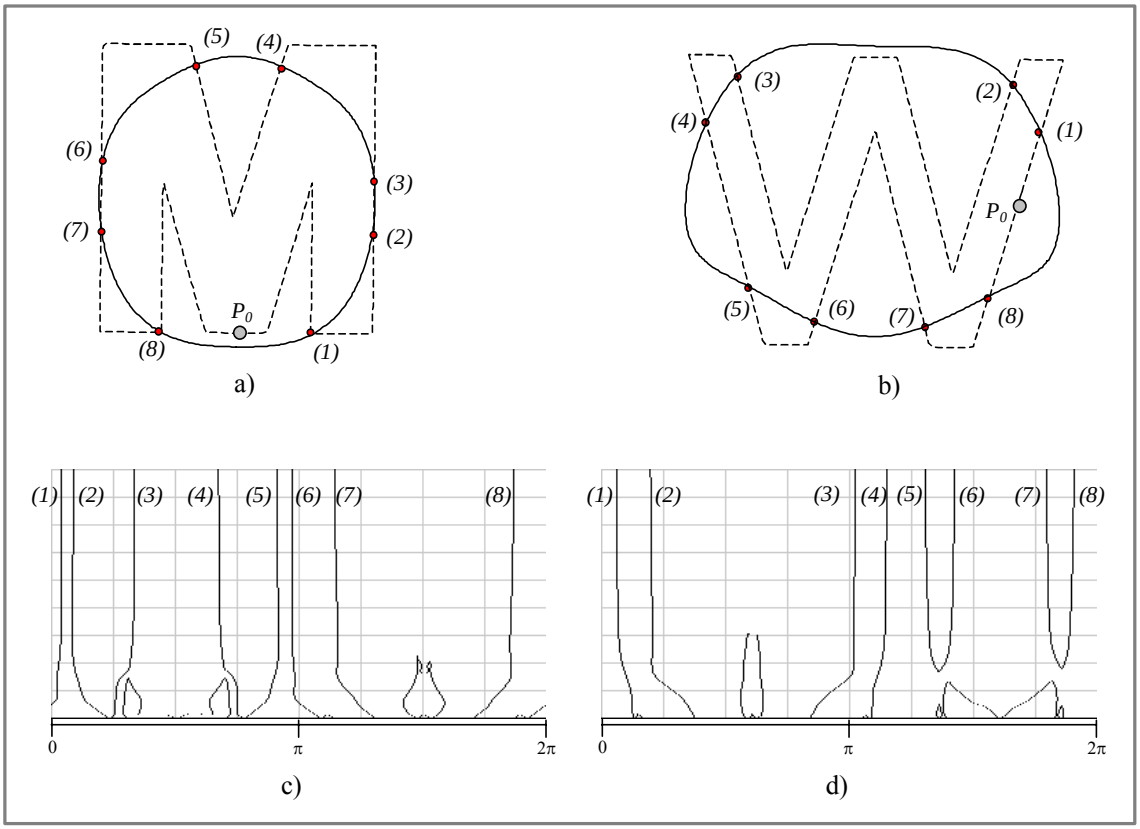


Fig.8:

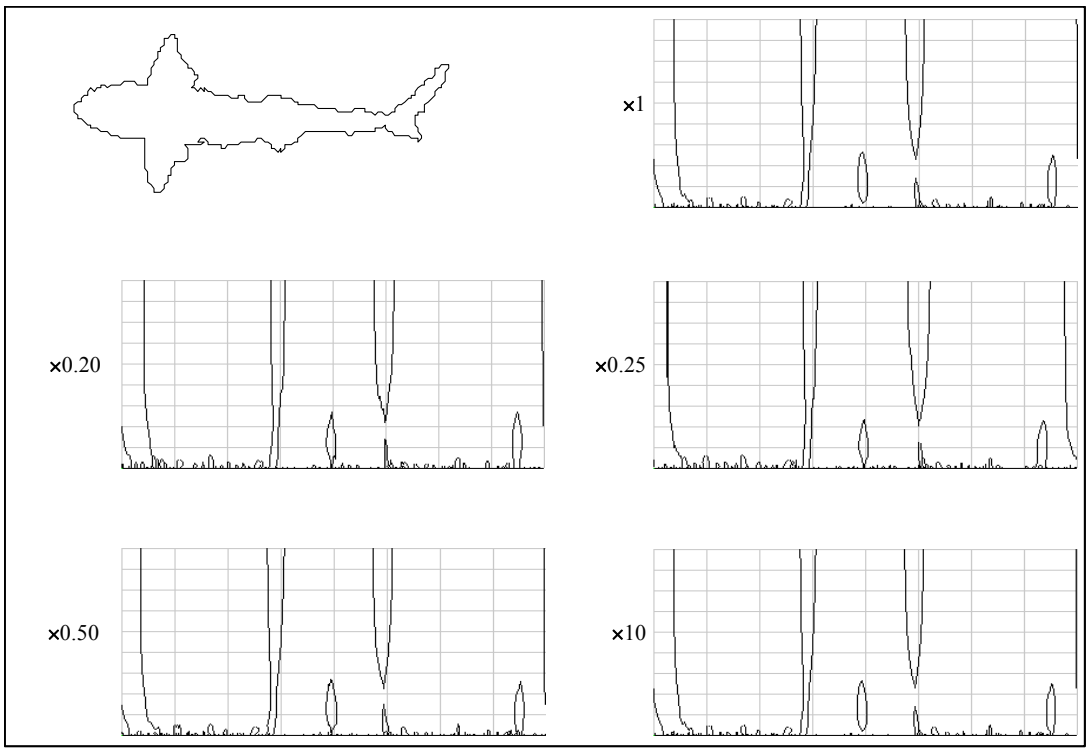


Fig.9:

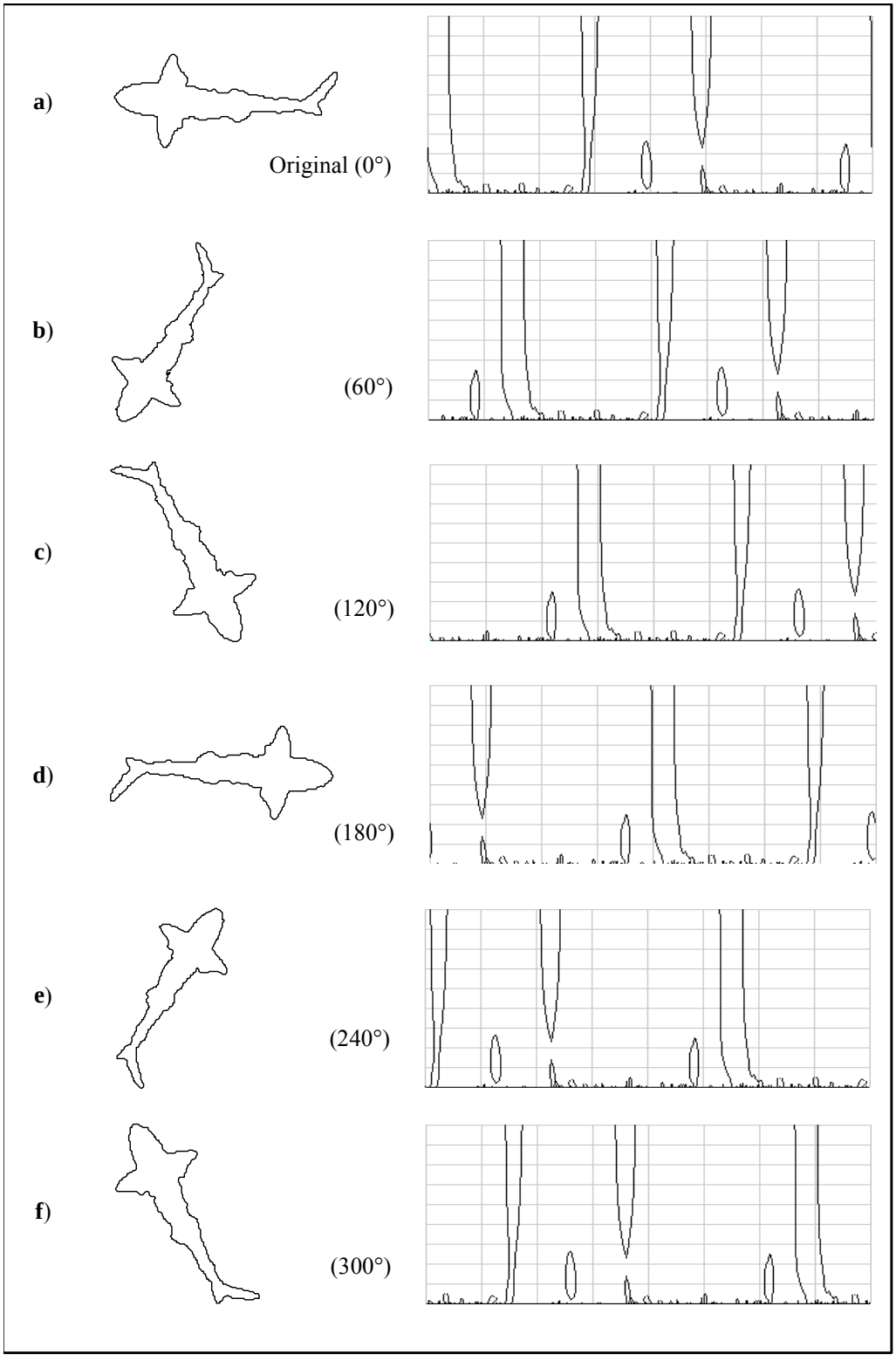


Fig.10:

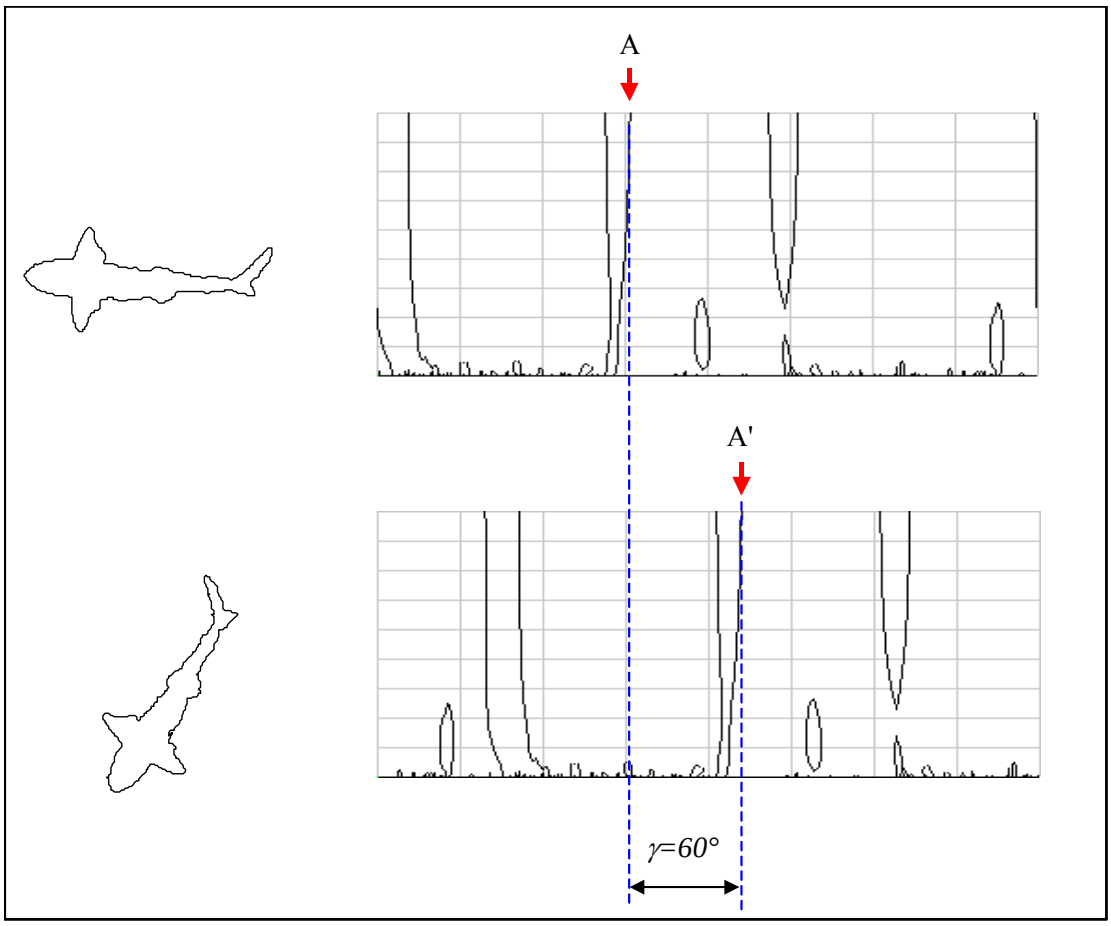


Fig.11:

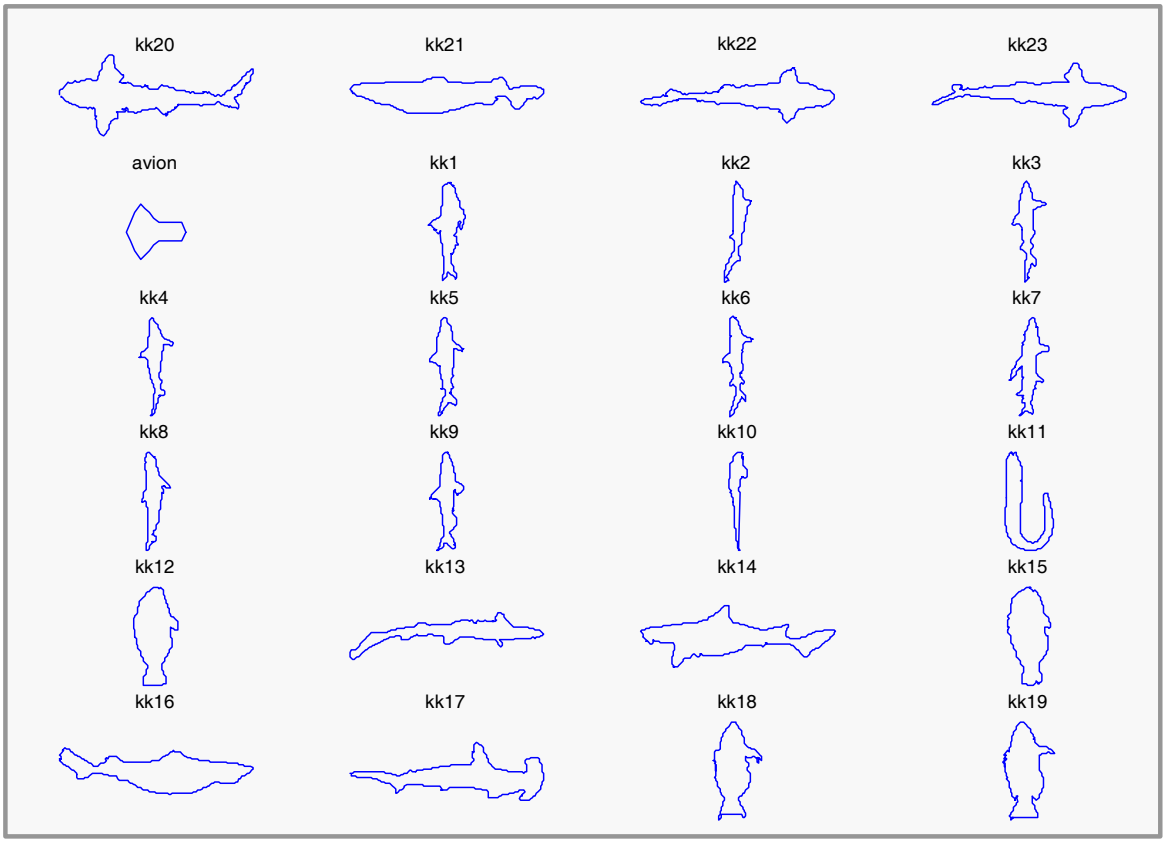

kk20
kk21
kk22
kk23
avion
kk1
kk2
kk3
kk4
kk5
kk6
kk7
kk8
kk9
kk10
kk11
kk12
kk13
kk14
kk15
kk16
kk17
kk18
kk19

Fig.12:

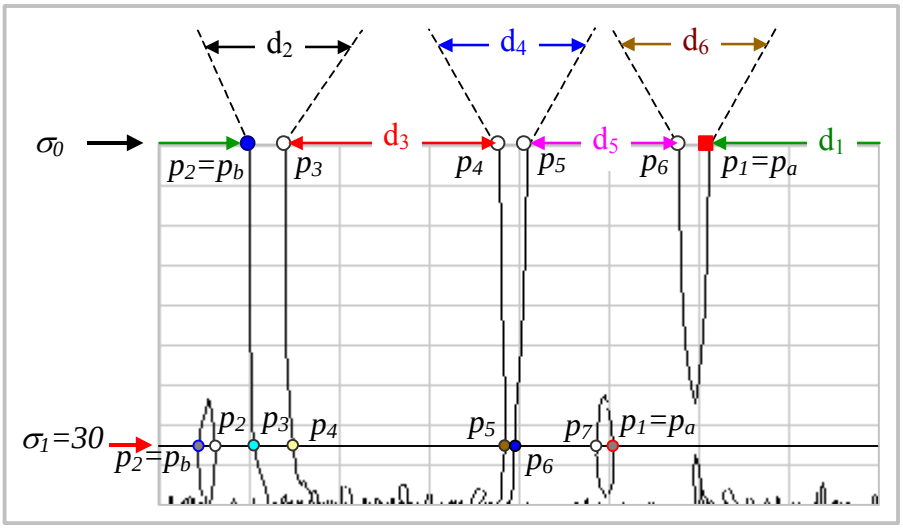

d2
d4
d6
σ0
d3
d5
d1
p2=pb
p3
p4
p5
p6
p1=pa
σ1=30
p2
p3
p4
p5
p7
p1=pa
p2=pb
p6

Fig.13:

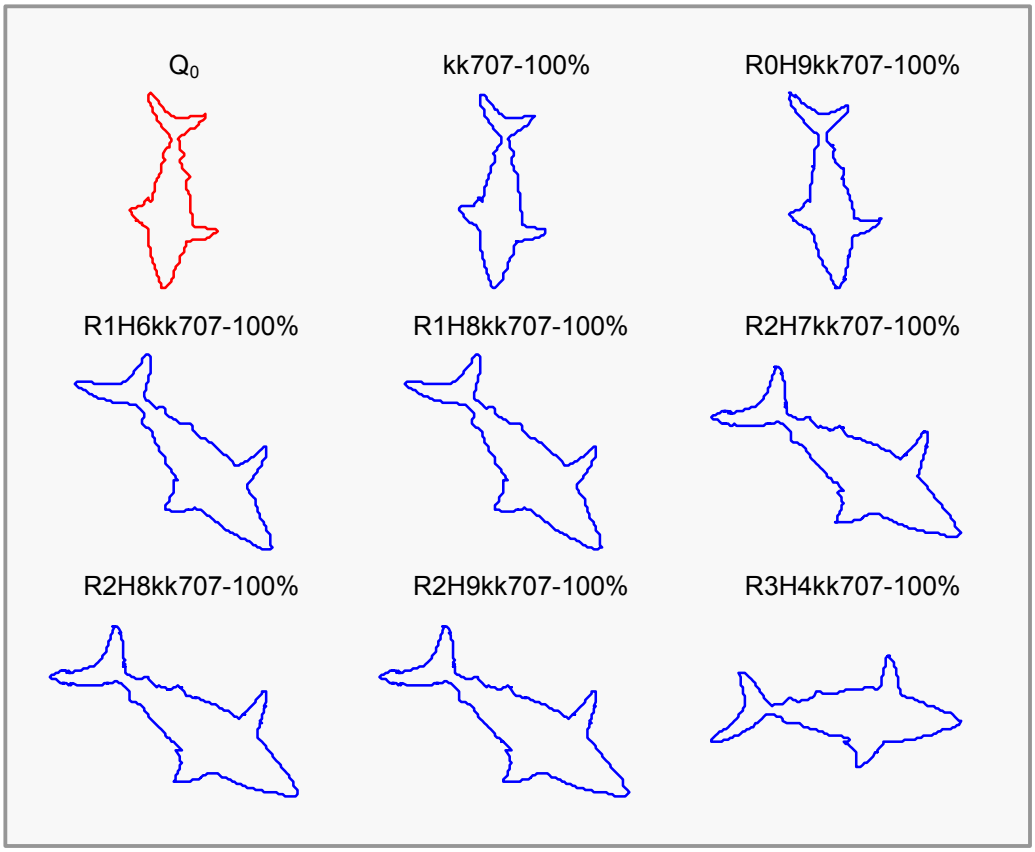

Q_0
kk707-100%
R0H9kk707-100%
R1H6kk707-100%
R1H8kk707-100%
R2H7kk707-100%
R2H8kk707-100%
R2H9kk707-100%
R3H4kk707-100%

Fig.14:

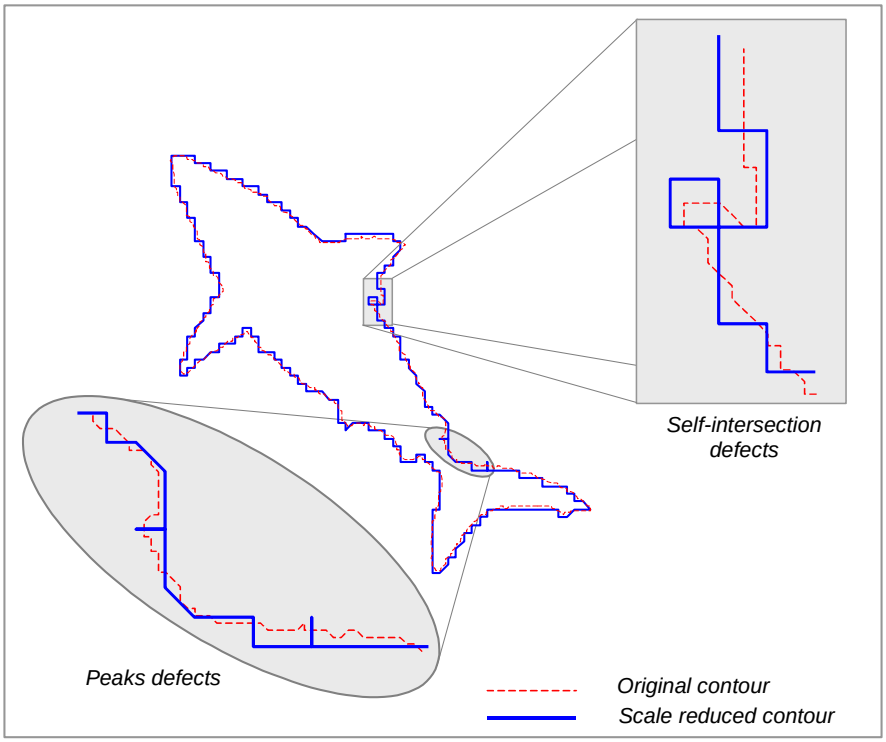
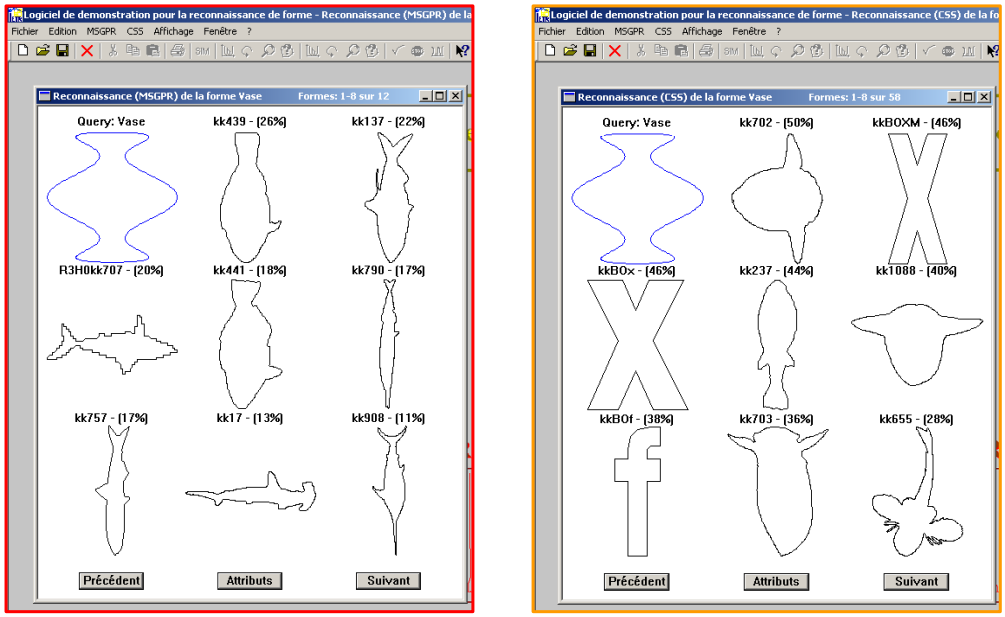
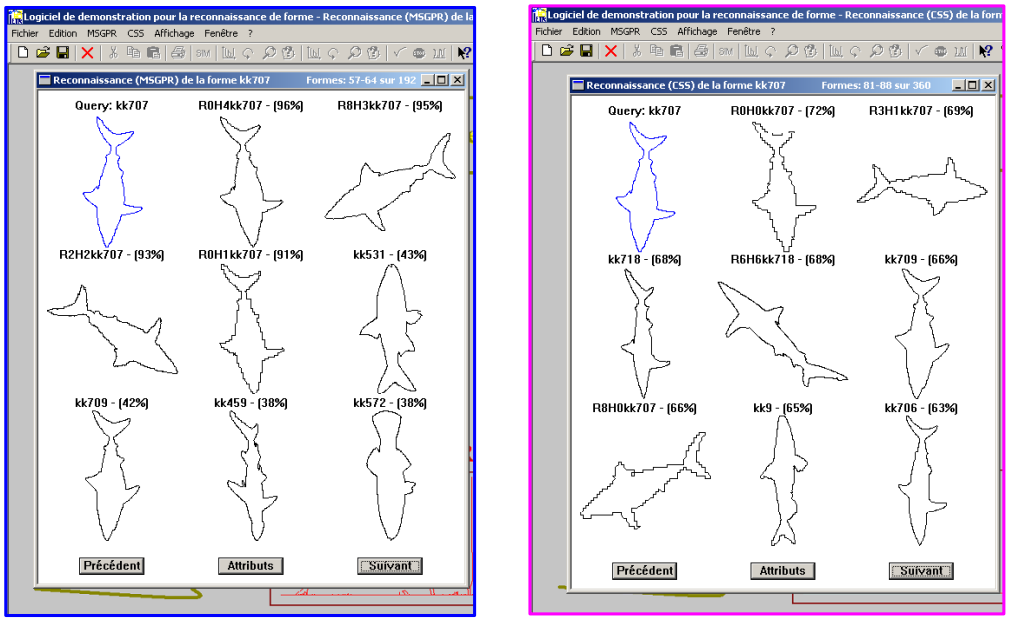

Self-intersection defects
Peaks defects
Original contour
Scale reduced contour

Fig.15:


Logiciel de demonstration pour la reconnaissance de forme - Reconnaissance (MSGPR) de la
Fichier Edition MSGPR CSS Affichage Fenêtre ?
Reconnaissance (MSGPR) de la forme Vase
Formes: 1-8 sur 12
Query: Vase
kk439 - (26%)
kk137 - (22%)
R3H0kk707 - (20%)
kk441 - (18%)
kk790 - (17%)
kk757 - (17%)
kk17 - (13%)
kk908 - (11%)
Précédent
Attributs
Suivant
Logiciel de demonstration pour la reconnaissance de forme - Reconnaissance (CSS) de la fo
Fichier Edition MSGPR CSS Affichage Fenêtre ?
Reconnaissance (CSS) de la forme Vase
Formes: 1-8 sur 58
Query: Vase
kk702 - (50%)
kkBOXM - (46%)
kkBOx - (46%)
kk237 - (44%)
kk1088 - (40%)
kkBOf - (38%)
kk703 - (36%)
kk655 - (28%)
Précédent
Attributs
Suivant

a) b)

Fig.16:


Logiciel de demonstration pour la reconnaissance de forme - Reconnaissance (MSGPR) de la
Fichier Edition MSGPR CSS Affichage Fenêtre ?
Reconnaissance (MSGPR) de la forme kk707
Formes: 57-64 sur 192
Query: kk707
R0H4kk707 - (96%)
R8H3kk707 - (95%)
R2H2kk707 - (93%)
R0H1kk707 - (91%)
kk531 - (43%)
kk709 - (42%)
kk459 - (38%)
kk572 - (38%)
Précédent
Attributs
Suivant
Logiciel de demonstration pour la reconnaissance de forme - Reconnaissance (CSS) de la form
Fichier Edition MSGPR CSS Affichage Fenêtre ?
Reconnaissance (CSS) de la forme kk707
Formes: 81-88 sur 360
Query: kk707
R0H0kk707 - (72%)
R3H1kk707 - (69%)
kk718 - (68%)
R6H6kk718 - (68%)
kk709 - (66%)
R8H0kk707 - (66%)
kk9 - (65%)
kk706 - (63%)
Précédent
Attributs
Suivant

a) b)

Fig.17:

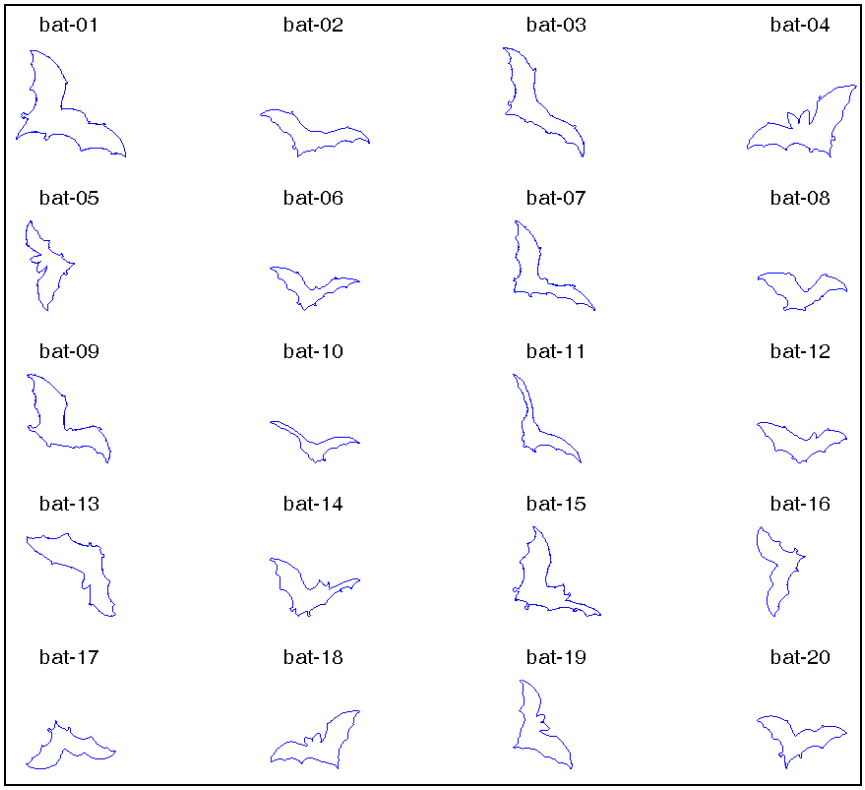

bat-01
bat-02
bat-03
bat-04
bat-05
bat-06
bat-07
bat-08
bat-09
bat-10
bat-11
bat-12
bat-13
bat-14
bat-15
bat-16
bat-17
bat-18
bat-19
bat-20

Fig.18:

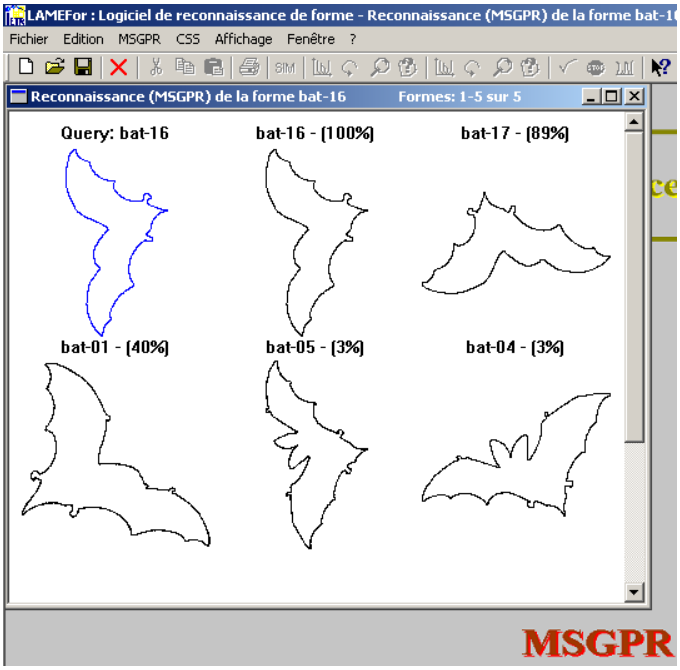

LAMEFor : Logiciel de reconnaissance de forme - Reconnaissance (MSGPR) de la forme bat-1
Fichier Edition MSGPR CSS Affichage Fenêtre ?
Reconnaissance (MSGPR) de la forme bat-16
Formes: 1-5 sur 5
Query: bat-16
bat-16 - (100%)
bat-17 - (89%)
bat-01 - (40%)
bat-05 - (3%)
bat-04 - (3%)
MSGPR

Fig.19:

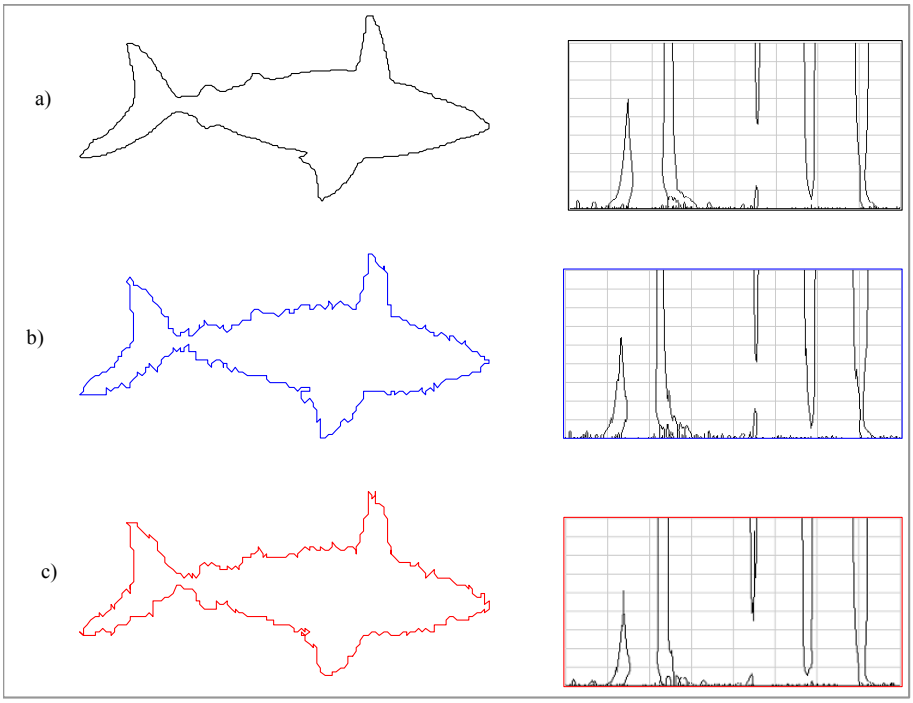


Table captions

Table I: Rotation angles and scaling factors used in affine transformations

Table II: Similarity scores for different contours in pattern retrieval from a database

Table III: Comparative results between IPM-based features and CSS-based ones in pattern retrieval.

Table I

Rotation angle (R_i)	10°	45°	60°	90°	135°	180°	225°	270°	300°	330°
Scaling factor (H_i)	0.2	0.25	0.5	0.75	1	2	25	10	20	30
Index i	0	1	2	3	4	5	6	7	8	9

Table II

kk707 (100)	R0H9kk707 (100)	R1H6kk707 (100)	R1H8kk707 (100)
R2H7kk707 (100)	**R2H8kk707 (100)**	**R2H9kk707 (100)**	**R3H4kk707 (100)**
R3H5kk707 (100)	R3H6kk707 (100)	R3H7kk707 (100)	R3H8kk707 (100)
R3H9kk707 (100)	**R4H7kk707 (100)**	**R5H4kk707 (100)**	**R5H5kk707 (100)**
R5H6kk707 (100)	R5H7kk707 (100)	R5H8kk707 (100)	R5H9kk707 (100)
R6H6kk707 (100)	**R6H9kk707 (100)**	**R7H4kk707 (100)**	**R7H5kk707 (100)**
R7H6kk707 (100)	R7H7kk707 (100)	R7H8kk707 (100)	R7H9kk707 (100)
R8H8kk707 (100)	**R8H9kk707 (100)**	**R9H8kk707 (100)**	**R9H9kk707 (100)**
R0H6kk707 (98)	R1H9kk707 (98)	R4H6kk707 (98)	R8H7kk707 (98)
R9H6kk707 (98)	**R0H8kk707 (98)**	**R6H7kk707 (98)**	**R6H8kk707 (98)**
R4H9kk707 (97)	R8H6kk707 (97)	R2H6kk707 (95)	R4H8kk707 (95)
R1H7kk707 (93)	**R2H5kk707 (87)**	**R9H7kk707 (87)**	**R8H5kk707 (85)**
R7H3kk707 (76)	R9H5kk707 (74)	R6H2kk707 (70)	R4H5kk707 (66)
R0H5kk707 (62)	**R8H4kk707 (55)**	**R6H4kk707 (54)**	**R3H2kk707 (52)**
R5H2kk707 (47)	R2H2kk707 (45)	R1H2kk707 (29)	R8H1kk707 (28)
R5H0kk707 (16)	**kk1097 (5)**	**R8H0kk707 (4)**	**kk882 (4)**
kk333 (3)	kk850 (2)	kk929 (2)	kk902 (2)
kk829 (2)	**R6H1kk707 (2)**	**kk511 (1)**	**kk916 (1)**

Table III

Method	Pattern	Average value of minimum similarity score	Average value of maximum similarity score	Average number of retained patterns
IPM	Bats	36.66	100.00	3.90
	Bell	10.85	100.00	20.00
	Children	11.55	100.00	20.00
	Global averages	**19.77**	**100.00**	**14.63**
CSS	Bats	50.93	100.00	4.60
	Bell	42.52	100.00	4.50
	Children	48.17	100.00	10.70
	Global average	**47.20**	**100.00**	**6.60**